Integration of Heterogeneous Distributed Database Based on E-Commerce System

Vinit Nimkar, Kazi FakirMohammed, Sushopti Gawde

***Abstract*—The rapid expansion of internet systems, ease of availability and so on reduction in operating costs, and spread of internet access have created colossal of electronic data. The disseminate nature of our internet system and databases, however, has resulted in a hodgepodge of diverse, heterogeneous, database implementations, making access to and gathering and combining of data across databases very difficult. The diacritic technologies used to make the integration are familiar. But our objective here it is the inaugural description of these technologies are presented together in well-organized to give a synopsis of how it is probable to collaborate different internet systems with purport relaxed ACID properties.**

The ACID (Atomicity, Consistency, Isolation and Durability) properties of a database are properties used (Data Base Management System) to make database recovery easier and make it possible in a multi user environment to give concurrent transactions a consistent chronological view of the data in the database. Therefore, the ACID properties are important. However in case of heterogeneous databases and there integration, the ACID properties may slow down the performance and the response time of the system. In many situations, the performance and the response time will be unacceptable if the ACID properties are implied on the heterogeneous databases. Hence in many real time application or business standards slow response and performance is not acceptable and needs to be optimized. Consequently, we will in this paper use relaxed ACID properties across different database. The objective of designing relaxed ACID properties across different database are to give the users a perspective that the data that may be inconsistent across various database but in practice functions as if the traditional ACID properties were implemented so that users can be assured about the data they are using.

In this paper, we will collaborate the ERP systems of the customers of an e-commerce website with the e-commerce system of the suppliers as an example. Despite the fact, the integration architecture or the conceptual design model proposed in this paper may for example also be used for collaborating airline systems, hospitality system, educational systems etc.

***Index Terms*—Adaptability, e-Learning, Bayesian Network., Learning Navigation.**

I. INTRODUCTION

A. Relaxed ACID properties:

The Atomicity property of a DBMS guarantees that either all the updates of a transaction are committed/executed or no updates are committed/executed. This property makes it possible to re-execute a transaction that has failed after execution of some of its updates. The Atomicity property of a DBMS is implemented by using a DBMS log file with all the database changes made by the transactions. The global Atomicity property of distributed databases with relaxed ACID properties is implemented by using compensatable, pivot and retriable sub transactions in that order as explained. By applying these sub transactions it is allowed to commit/execute only part of the transaction and still consider the transaction to be atomic.

The global Consistency property is not defined in databases with relaxed ACID properties because normally such databases are inconsistent and this inconsistency may be managed in the same way as the relaxed Isolation property.

The Isolation property of a DBMS guarantees that the updates of a transaction cannot be seen by other concurrent transactions until the transaction is committed/executed. That is the inconsistencies cause by a transaction that has not executed all its updates cannot be seen by other transactions. The Isolation property of a DBMS may be implemented by locking all records used by a transaction. That is the locked records cannot be used by other transactions before the locks are released when the transaction is committed. The global Isolation property of databases with relaxed ACID properties is implemented by using countermeasures against the inconsistencies/anomalies that may occur. This is explained in more details further section.

The Durability property of a DBMS guarantees that the updates of a transaction cannot be lost if the transaction is committed. The Durability property of a DBMS is implemented by using a DBMS log file with all the database changes made by the transactions. By restoring the updates of the committed transactions it is possible to recover a database even in case it is destroyed. The global Durability property of databases with relaxed ACID properties is implemented by using the local Durability property of the local databases involved.

B. Objective:

Class ERP systems consist of integrated local modules like e-commerce and traditional sale, procurement, production management, logistics management, CRM (Customer

Manuscript received January 08, 2015

Vinit Nimkar, Information Technology, Pillai's Institute of Information Technology, Mumbai, India, Mob No: 9773294378

Kazi FakirMohammed, Information Technology, Pillai's Institute of Information Technology, Mumbai, India, Mob No: 90004033643

Sushopti Gawde Information Technology, Pillai's Institute of Information Technology, Mumbai, India,Mob No: 9920730728

Relations Management), HRM (Human Resource Management), etc. The objective in this paper is to describe a model for how it is possible to integrate such systems/modules in a consistent way across different locations by using SOA services.

II. TRANSACTION MODEL

A multidatabase is a union of local autonomous databases. Global transactions access data located in more than one local database. In recent years, many transaction models have been designed to integrate local databases without using a distributed DBMS. The countermeasure transaction model has, among other things, selected and integrated properties from these transaction models to reduce the problems, as reliability and consistency, caused by the missing ACID properties in a distributed database that is not managed by a distributed DBMS. In the countermeasure transaction model, a global transaction involves a root transaction (client transaction) and several single site subtransactions (server transactions). Subtransactions may be nested transactions, i.e. a subtransaction may be a parent transaction for other subtransactions. All communication with the user is managed from the root transaction, and all data is accessed through subtransactions. The following subsections will give a broad outline of how relaxed ACID properties have to be implemented.

A. Atomicityproperty:

Adaptive e-Learning system prototype (Picture 1) can be generally divided into 3 layers: concept layer, relation layer and user view layer.

Pivot subtransaction: It manages the atomicity of the global transactions. A global transaction is said to be committed when the pivot subtransaction is committed locally. Therefore, a global transaction can only have one pivot subtransaction. If the pivot subtransaction aborts, all updates of other subtransactions must be compensated/removed in order to obtain the atomicity property.

Compensatable subtransaction: The compensatable subtransactions ensure the possibility for compensation. Compensatable subtransactions must always be executed before the pivot subtransaction is executed to make it possible to compensate them if the pivot subtransaction cannot be committed. A compensatable subtransaction may be compensated by executing a so-called compensating subtransaction that removes the database changes made by the corresponding compensatable subtransaction. In integrated ERP systems, we will recommend to use compensatable subtransactions to decrease stocks levels in other locations and retriable subtransactions to increase stocks levels in otherlocations in order to make it impossible for the users to use non existing stocks.

Retriable subtransaction: The retriable subtransactions that are designed in such a way that the execution is guaranteed to commit locally (sooner or later) if the pivot subtransaction has been committed. Therefore, it is possible (without violating the atomicity property) to execute retriable subtransactions after the global transaction is committed by the pivot subtransaction. Retriable subtransactions may for example be used to update replicated data, execute compensating subtransactions, or increase stock levels as described above.

The global atomicity property is implemented by executing the compensatable, pivot and retriable subtransactions of a global transaction in that order. For example, if the global transaction fails before the pivot has been committed, it is possible to remove the updates of the global transaction by compensation. If the global transaction fails after the pivot has been committed, the remaining retriable subtransactions will be (re)executed automatically until all the updates of the global transaction have been committed.

B. Consistency property:

A database is consistent if its data complies with the consistency rules of the database. If the database is consistent both when a transaction starts and when it has been completed and committed, the execution has the consistency property. Transaction consistency rules may be implemented as a control program that rejects the commitment of transactions, which do not comply with the consistency rules.

The above definition of the consistency property is not useful in distributed databases with relaxed ACID properties because such a database is almost always inconsistent. However, a distributed database with relaxed ACID properties should have asymptotic consistency, i.e. the database should converge towards a consistent state when all active transactions have been committed/compensated. Therefore, the following property is essential in distributed databases with relaxed ACID properties:

If the database is asymptotically consistent when a transaction starts and also when it has been committed, the execution has the relaxed consistency property.

C. Isolation property:

For The isolation property is normally implemented by using long duration locks, which are locks that are held until the global transaction has been committed. In the countermeasure transaction model, long duration locks cannot instigate isolated global execution as retriable subtransactions may be executed after the global transaction has been committed in the pivot location. Therefore, short duration locks are used, i.e. locks that are released immediately after a subtransaction has been committed/aborted locally. To ensure high availability in locked data, short duration locks should also be used in compensatable subtransactions, just as locks should be released before interaction with a user. This is not a problem in the countermeasure transaction model as the traditional isolation property in retriable subtransactions is lost anyway. If only short duration locks are used, it is impossible to block data. (Data is blocked if it is locked by a subtransaction that loses the connection to the “coordinator” (the pivot subtransaction) managing the global commit/abort decision). When transactions are executed without isolation, the so-called isolation anomalies may occur. The countermeasure transaction model describes countermeasures that reduce the problems of the anomalies.

D. *Durability*:

Updates of transactions are said to be durable if they are stored in a stable manner and secured by a log recovery system. In case a global transaction has the atomicity property (or relaxed atomicity), the global durability property (or relaxed durability property) will automatically be implemented, as it is ensured by the log-system of the local DBMS systems.

III. DISTRIBUTED COMMUNICATION BETWEEN DIFFERENT TRANSACTION MODEL

In order to implement integration flexibility (fault tolerance in case a backup unit is used) between the ERP modules of different companies, it is not acceptable that the different modules communicate directly with each other. All communication between different modules must be executed by applications offered as e.g. SOA services. In order also to implement relaxed ACID properties between the modules each module should offer the following types of services:

Therefore, the different ERP modules should offer the following types of services as these are the basis for implementing the Atomicity property across different ERP locations:

Read only services that are used when a local internet system wants to read data managed by another local internet system. Compensatable update services that are used when a local internet system wants to make compensatable updates in tables managed by another local internet system.

Retriable update services that are used when a local internet system wants to make retriable updates in tables managed by another local internet system. By using these three types of services from other local internet systems it should be possible for any local internet system to make distributedupdates with relaxed ACID properties.

IV. CONCEPTUAL DESIGN MODEL

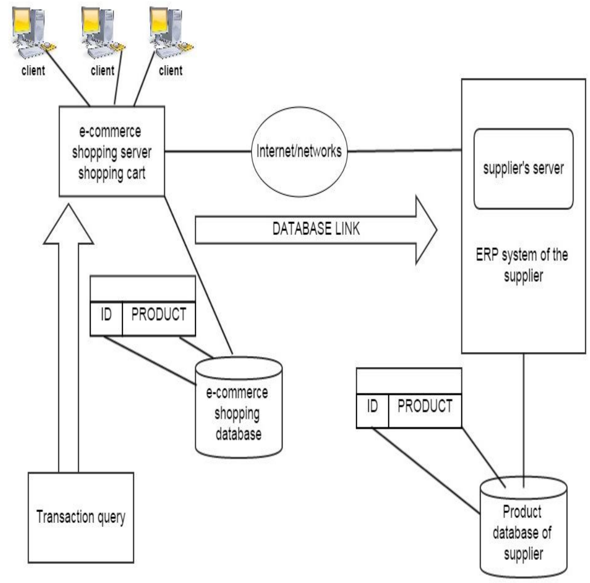
client
client
client
e-commerce shopping server shopping cart
Internet/networks
supplier's server
ERP system of the supplier
DATABASE LINK
ID
PRODUCT
e-commerce shopping database
ID
PRODUCT
Transaction query
Product database of supplier

Fig 1. Conceptual Design Model

Dynamic website will be created in the front end of the e-commerce with the help of jsp (java server pages).AJAX will be used in the background so that changes can be made in the database without refreshing web page when products are added or removed from the virtual cart. MYSQL databases are used so that the two databases i.e.e-commerce shopping database and the product database of the supplier can be linked with each other. Product ID will be kept as the primary key and foreign key. If theProduct tally becomes less than threshold value of the product stock a query will be fired to check the product stocks in the joined databases. If the product is available in the supplier database than customer is permitted to go ahead with the order otherwise is asked to modify the order if he doesn't wish to change the order he can cancel his order.

V. EXAMPLE

Common for most e-commerce systems is that when a customer wants to make a new order, the customer first puts the needed products in his virtual basket. In nonintegrated e-commerce systems, the updates of the product stocks are executed after the customer has finished his order. In ERP integrated e-commerce systems, the updates of the product stocks of the owner of the e-commerce system may be executed after each new type of product put into the basket of the customer. In this case it is not important that the integration is implemented by using web-services because most enterprises select an e -commerce system that is prepared for integration with their ERP system. However, this is not always possible when e-commerce systems are going to be integrated with the ERP systems of the suppliers as the suppliers may change.In this paper we will define a supplier integrated e-commerce systems, as e-commerce systems where it is possible for a customer to order products directly from the stocks of different suppliersthat is the e-commerce system must be integrated with the ERP systems of the different suppliers. For both ERP integrated e-commerce systems and supplier integrated e-commerce systems integration may be implemented by using read only, compensatable, and retriable web-services in the following way:

For each new product put into the basket a compensatable web-service is used to executes a compensatable decrease in the stock location If the e-commerce system receives the answers that the compensatable decreases have been committed locally for all the products in the order, the e-commerce system can commit the order globally in its own location On the other hand, if the e-commerce system receives the answer that a compensatable service cannot be committed, the e-commerce system must ask the customer to change the order.If the customer cannot change the order in a proper way the e-commerce system must abort the global transaction by executing a retriable compensating transaction in the locations, where the amount of stocks have been updated.

VI. CONCLUSION

In the theoretical part of this paper, we have described how it is possible to integrate interne systems/web-databases by using read only, compensatable, and retriable web-services to implement relaxed ACID properties across the different web-databases. The practical example illustrates how it is possible to integrate e-commerce systems with the ERP systems of the suppliers. Many other types of industry may benefit by using the integration theory described in this paper and therefore, it is our objective to analyzeand describe more of these possibilities in the future.

REFERENCES:

[1] Breibart, Y., Garcia-Molina, H. and Silberschatz, A., 1992, “Overview of Multidatabase Transaction Management”, VLDB Journal, 2, pp 181-239.

[2] Berenson, H., Bernstein, P., Gray, J., Melton, J., O’Neil, E. and O’Neil, P., 1995, “A Critique of ANSI SQL Isolation Levels”, Proc ACM SIGMOD Conf., pp. 1-10.

[3]Frank, L, 2010, Design of Distributed Integrated Heterogeneous or Mobile Databases, LAP LAMBERT Academic Publishing AG & Co. KG, Germany, ISBN 978-3-8383-4426-3, pp 1-157.

[4]Frank, L. and Zahle, T, 1998, “Semantic ACID Properties in Multidatabases Using Remote Procedure Calls and Update Propagations”, Software - Practice & Experience, Vol.28, pp77-98.

[5]Gray, J. and Reuter, A., 1993, “Transaction Processing”, Morgan Kaufman, 1993.

[6]Garcia-Molina, H. and Salem, K., 1987, “Sagas”, ACM SIGMOD Conf, pp 249-259.

[7]Mehrotra, S., Rastogi, R., Korth, H., and Silberschatz, A., 1992, “A transaction model for multi-database systems”, Proc International Conference on Distributed Computing Systems, pp 56-63.

[8]Weikum, G. and Schek, H., 1992, “Concepts and Applications of Multilevel Transactions and Open Nested Transactions”, A. Elmagarmid (ed.): Database Transaction Models for Advanced Applications, Morgan Kaufmann, pp 515-553.

[9]Zhang, A., Nodine, M., Bhargava, B. and Bukhres, O., 1994, “Ensuring Relaxed Atomicity for Flexible Transactions in Multidatabase Systems”, Proc ACM SIGMOD Conf, pp 67-78.

VINIT NIMKAR, Masters of Engineering, Student, PIIT, Panvel Navi Mumbai. Area of Interest: Database.

KAZI FAKIRMOHAMMED, Masters of Engineering, Student, PIIT, Panvel, Navi Mumbai. Won 1st Prize in National Level Papers Presentation QUEST ’06 for the Paper Nano Technology. Area of Interest: Database.

SUSHOPTI GAWDE has Bachelors and Master's Degree in the field of Computer science and Engineering. She has keen interest in the area of Web Engineering, Usability Engineering, Image Processing, Ubiquitous Computing, E-learning, E-commerce, and has published several papers in National and International conferences and journals. Her interest is also extended to data storage and information retrieval. Prof. Sushopti has more than fourteen years of experience in teaching undergraduate as well as postgraduate students. She has attended several workshops and faculty development programs. She has 9 Research Papers to her credit.